On Geometric Set Cover for Orthants

Karl Bringmann
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
kbringma@mpi-inf.mpg.de

Sándor Kisfaludi-Bak
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
s.kisfaludi.bak@tue.nl

Michał Pilipczuk
Institute of Informatics, University of Warsaw, Poland
michal.pilipczuk@mimuw.edu.pl

Erik Jan van Leeuwen
Dept. Information & Computing Sciences, Utrecht University, Utrecht, The Netherlands
e.j.vanleeuwen@uu.nl

Abstract

We study Set Cover for orthants: Given a set of points in a d-dimensional Euclidean space and a set of orthants of the form $(-\infty, p_1] \times \ldots \times (-\infty, p_d]$, select a minimum number of orthants so that every point is contained in at least one selected orthant. This problem draws its motivation from applications in multi-objective optimization problems. While for $d = 2$ the problem can be solved in polynomial time, for $d > 2$ no algorithm is known that avoids the enumeration of all size-k subsets of the input to test whether there is a set cover of size k. Our contribution is a precise understanding of the complexity of this problem in any dimension $d \geqslant 3$, when k is considered a parameter:

- For $d = 3$, we give an algorithm with runtime $n^{\mathcal{O}(\sqrt{k})}$, thus avoiding exhaustive enumeration.
- For $d = 3$, we prove a tight lower bound of $n^{\Omega(\sqrt{k})}$ (assuming ETH).
- For $d \geqslant 4$, we prove a tight lower bound of $n^{\Omega(k)}$ (assuming ETH).

Here n is the size of the set of points plus the size of the set of orthants. The first statement comes as a corollary of a more general result: an algorithm for Set Cover for half-spaces in dimension 3. In particular, we show that given a set of points U in $\mathbb{R}^3$, a set of half-spaces $\mathcal{D}$ in $\mathbb{R}^3$, and an integer k, one can decide whether U can be covered by the union of at most k half-spaces from $\mathcal{D}$ in time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot |U|^{\mathcal{O}(1)}$.

We also study approximation for Set Cover for orthants. While in dimension 3 a PTAS can be inferred from existing results, we show that in dimension 4 and larger, there is no 1.05-approximation algorithm with runtime $f(k) \cdot n^{o(k)}$ for any computable f, where k is the optimum.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Computational complexity and cryptography

Keywords and phrases Set Cover, parameterized complexity, algorithms, Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.26

Funding *Sándor Kisfaludi-Bak*: Supported by the Netherlands Organization for Scientific Research NWO under project no. 024.002.003.
Michał Pilipczuk: This work is a part of project TOTAL (Mi. Pilipczuk) that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 677651).

Acknowledgements We are grateful to an anonymous reviewer who pointed to us the reduction of Pach and Tardos [46] from Orthant Cover to Geometric Set Cover for half-spaces, and who suggested taking a closer look at the case of Geometric Set Cover for half-spaces. We would like to also thank the reviewer for pointing out that a PTAS for Orthant Cover in dimension 3 follows from the work of Mustafa and Ray [44] composed with the reduction of Pach and Tardos [46], which replaced our previous ad-hoc argument. We thank the organizers of the workshop on fixed-parameter computational geometry, held in May 2018 at Lorentz Center in Leiden, the Netherlands, where the main conceptual part of this work was done.

1 Introduction

Motivated by applications in multi-objective optimization, we study a geometric variant of the classic Set Cover problem. In general, Set Cover is defined as follows. Let $\mathbb{U}$ be a universe; typically, $\mathbb{U}$ is a finite collection of elements or $\mathbb{R}^d$ for some constant $d \geqslant 1$. Given a finite set $U \subseteq \mathbb{U}$ and a finite set $\mathcal{T}$ of subsets of $\mathbb{U}$, the goal is to find a set $S \subseteq \mathcal{T}$ of minimum size such that for each $u \in U$ it holds that $u \in F$ for some $F \in S$. We let $n = |\mathcal{T}| + |U|$.

Set Cover can be approximated within factor $\ln |\mathcal{T}|$ by a greedy algorithm [13, 29, 34], but, unless P = NP, no polynomial-time algorithm can attain an approximation factor of $(1 - \varepsilon) \ln |\mathcal{T}|$ for any $\varepsilon > 0$ [20]. Moreover, when parameterized by the expected solution size k (formally, here we consider a decision problem), the problem is W[2]-hard [21] and there is no $\mathcal{O}(n^{k-\varepsilon})$-time algorithm for any $\varepsilon > 0$, unless the Strong Exponential Time Hypothesis (SETH) fails [50]. Recently, it was even shown that, unless the Gap Exponential Time Hypothesis (Gap-ETH) fails, Set Cover has no $f(\mathrm{OPT})|\mathcal{T}|^{o(|\mathrm{OPT}|)}$-time algorithm that approximates the optimum OPT within a factor of $a(\mathrm{OPT})$, for any computable a and f [9]. This makes Set Cover a very hard algorithmic problem in general.

Fortunately, through years of research, we know that Set Cover becomes much easier when geometry is involved. If the universe $\mathbb{U}$ is equal to $\mathbb{R}^d$ for some $d \geqslant 1$, the set U is a set of points, and the sets in $\mathcal{T}$ are defined by geometric objects, then the problem is known as Geometric Set Cover. Then various restrictions on the shapes of objects allowed in $\mathcal{T}$ may lead to different tractability results. While for $d = 1$ the problem is polynomial-time solvable when $\mathcal{T}$ is required to consist of intervals, there are easy cases in $d = 2$ that are NP-hard, such as when $\mathcal{T}$ is defined by sets of unit squares or disks [24, 30]. However, the approximability of Geometric Set Cover in $d = 2$ is significantly better than in general. Approaches like the shifting technique [23], ε-nets [1, 2, 8, 14, 33, 41, 47], local search [3, 25, 44], sampling techniques [11, 55] and separator techniques [43] have proven successful in obtaining constant-factor approximation algorithms and approximation schemes. Recently, Govindarajan et al. [25] showed a very general approximability result, namely that Geometric Set Cover admits a PTAS when the underlying sets in $\mathcal{T}$ are non-piercing regions, which includes the case of pseudo-disks. From a parameterized perspective, Marx and Pilipczuk [39] showed that Geometric Set Cover has a $|\mathcal{T}|^{\mathcal{O}(\sqrt{k})}$-time algorithm when $\mathcal{T}$ is a set of disks or a set of squares. Moreover, no $n^{o(\sqrt{k})}$-time algorithm exists for these cases unless the Exponential Time Hypothesis fails [36, 39]. For piercing regions, such as axis-parallel rectangles and fat triangles, Geometric Set Cover is APX-hard [10, 27] and admits no $|\mathcal{T}|^{o(k)}$-time algorithm unless ETH fails [39]. For $d = 3$, a generic PTAS is also unlikely, as Geometric Set Cover is APX-hard even for unit balls [10], although constant-factor approximation algorithms do exist in certain cases [33]. This makes the complexity of Geometric Set Cover highly interesting for $d \geqslant 3$.

Orthant Cover. In this paper, we contribute to the knowledge about Geometric Set Cover by considering the case when the sets in $\mathcal{T}$ are orthants, which we call Orthant Cover. An *orthant* is a subset $T \subset \mathbb{R}^d$ of the form $T = \{(x_1, \ldots, x_d) \in \mathbb{R}^d \colon x_i \leqslant p_i$ for all $i \in [d]\}$ for some $(p_1, \ldots, p_d) \in \mathbb{R}^d$. Alternatively, an orthant can be defined as $(-\infty, p_1] \times \cdots \times (-\infty, p_d]$.

Our interest in Orthant Cover is motivated by multi-objective optimization. Here an optimization problem (like shortest path) is associated with $d > 1$ objectives (e.g. every edge has a cost and transition time), see, e.g., [5, 22, 26, 45, 48, 49]. We identify each possible solution of the optimization problem with the vector in $\mathbb{R}^d$ that lists all of its d objectives. A solution $p \in \mathbb{R}^d$ is called *Pareto-optimal* if there is no solution $q \in \mathbb{R}^d$ with $q \geqslant p$ (i.e., $q_i \geqslant p_i$ for all $1 \leqslant i \leqslant d$). The set of all Pareto-optimal solutions $F \subseteq \mathbb{R}^d$ is called the *Pareto front* [32] (or *trade-off curve* [56] or *skyline* [4]), and computing it is the standard goal of multi-objective optimization.

However, the Pareto front can be prohibitively large to display to the end user. Therefore, a typical relaxation is to compute a $(1+\varepsilon)$-approximation of the Pareto front. This is defined as a subset F' of the Pareto front F such that for every $p \in F$ there exists a $q \in F'$ with $p \leqslant (1+\varepsilon)q$ [48]. The question then becomes to find a Pareto front approximation of minimum size. This problem has been studied in multi-objective optimization under different names like "approximately dominating representatives" (ADR) [32] and "ε-indicator subset selection" [6, 7, 57]. Observe that we can solve ADR using an algorithm for Orthant Cover by setting

$$U := F \qquad \text{and} \qquad \mathcal{T} := \{(-\infty,(1+\varepsilon)f_1] \times \ldots \times (-\infty,(1+\varepsilon)f_d] \ : \ (f_1,\ldots,f_d) \in F\}.$$

Therefore, Orthant Cover can be seen as an asymmetric variant of ADR. This provides strong motivation to gain an algorithmic understanding of Orthant Cover.

We already know that in dimension $d = 2$, Orthant Cover can be solved in polynomial time, and even in near-linear time in n [7, 32]. For $d \geqslant 3$, however, the problem becomes NP-hard [32]. Moreover, if we focus on looking for a solution of size at most k, no algorithm is known that avoids the enumeration of all size-k subsets of $\mathcal{T}$. In fact, no $n^{o(k)}$-time algorithm is known, even for $d = 3$. Therefore, we ask *in which dimensions can the naive algorithm for* Orthant Cover *with running time $n^{\mathcal{O}(k)}$ be significantly improved upon?*

Our Contribution. In this paper, we resolve the parameterized complexity of Orthant Cover when parameterized by the size of the solution. We present an algorithm for $d = 3$ that improves on the naive $n^{\mathcal{O}(k)}$-time algorithm, and rule out any further significant improvements in any dimension. Our lower bounds are conditional on the Exponential Time Hypothesis (ETH) by Impagliazzo, Paturi, and Zane [28], which (avoiding technical details) states that 3-SAT has no algorithm with running time $2^{o(n)}$, where n is the number of variables.

▶ **Theorem 1.** *Consider the* Orthant Cover *problem in dimension d. Then:*
1. *for $d = 3$, it can be solved in time $|\mathcal{T}|^{\mathcal{O}(\sqrt{k})} \cdot |U|^{\mathcal{O}(1)}$, in particular in time $n^{\mathcal{O}(\sqrt{k})}$;*
2. *for $d = 3$, it cannot be solved in time $f(k)\, n^{o(\sqrt{k})}$ for any computable f, assuming ETH;*
3. *for $d \geqslant 4$, it cannot be solved in time $f(k)\, n^{o(k)}$ for any computable f, assuming ETH.*

In the above and for all the results stated in this paper, we measure the running time in the number of arithmetic operations over the reals given on input, i.e., in the strong fashion. Note that $n = |\mathcal{T}| + |U|$.

Thus, we determine the optimal time complexity of Orthant Cover as $n^{\Theta(\sqrt{k})}$ for $d = 3$ and $n^{\Theta(k)}$ for $d \geqslant 4$, assuming ETH. This dependence on d is somewhat surprising, since many previous conditional lower bounds for geometric problems are of the form $n^{\Omega(k^{1-1/d})}$ [40, 52]. We are only aware of one other work establishing problems to be easier for $d = 3$, but for $d = 4$ to be as hard as in any high dimension, namely k-means and k-median [15].

The algorithm of Theorem 1.1 actually follows from a more general result.

▶ **Theorem 2.** *Given a set of points U in $\mathbb{R}^3$, a set of half-spaces $\mathcal{D}$ in $\mathbb{R}^3$, and an integer k, one can decide whether U can be covered by the union of at most k half-spaces from $\mathcal{D}$ in time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot |U|^{\mathcal{O}(1)}$.*

It is known that ORTHANT COVER can be reduced to this case (see [46, Lemma 2.3] or [12, Section A.3]). We observe that GEOMETRIC SET COVER for disks in $\mathbb{R}^2$ (DISK COVER) can also be reduced to this case, as follows. Consider an instance of DISK COVER where the point set U and the disk set $\mathcal{T}$ are in the plane $z = 0$, and let p be a point in $\mathbb{R}^3$ outside this plane. For each disk $D \in \mathcal{T}$, we can define a ball $B(D)$ whose intersection with the plane $z = 0$ is D and that has p on its boundary. We apply an inversion with center p. As a result, each ball $B(D)$ is mapped to a half-space that contains the inverse of a point $x \in U$ if and only if x is covered by D. Hence, Theorem 2 also generalizes the known $n^{\mathcal{O}(\sqrt{k})}$-time algorithm for DISK COVER [39].

We also study the approximability of ORTHANT COVER. Previous work implies a PTAS for $d = 3$ running in $n^{\mathcal{O}(1/\varepsilon^2)}$ time by a reduction (see [46, Lemma 2.3] or [12, Section A.3]) to the known PTAS for half-spaces in dimension 3 [44], and APX-hardness for $d \geqslant 4$ by a reduction (see Section 4) to the known APX-hardness of RECTANGLE COVER [54]. In this paper, we rule out any significant improvement for $d = 3$, particularly the existence of an Efficient PTAS. For $d \geqslant 4$, we establish a stronger inapproximability result conditional on Gap-ETH [19, 35].

▶ **Theorem 3.** *Consider the* ORTHANT COVER *problem in dimension d. Then:*
1. *for $d = 3$, it has no PTAS with running time $f(\varepsilon)\, n^{o(\sqrt{1/\varepsilon})}$ for any computable f, assuming ETH;*
2. *for any $d \geqslant 4$, it has no 1.05-approximation algorithm running in time $f(k)n^{o(k)}$ for any computable f, assuming Gap-ETH.*

Technical Overview. Our algorithm for half-spaces in $\mathcal{R}^3$ is a branching algorithm that attempts to split the input point set based on a balanced separator $\mathcal{S}_0$ of the optimum solution, where the separator should be small: of size $\mathcal{O}(\sqrt{k})$. However, we do not know the optimum solution and thus we cannot know the separator. Instead, we show that we can enumerate a set of candidate separators in time $|\mathcal{T}|^{\mathcal{O}(\sqrt{k})}$, in which the separator $\mathcal{S}_0$ is guaranteed to be contained. Similar approaches to obtain a subexponential-time algorithm for geometric and planar problems are known to exist (e.g. [31, 39]). However, the existence of the balanced separator of size $\mathcal{O}(\sqrt{k})$ is somewhat surprising here, since in 3 dimensions only separators of size $\mathcal{O}(k^{2/3})$ are known (see e.g. [53]). In order to get the desired separator size, we work on the surface of the convex polytope which is defined as the complement of the union of half-spaces in the solution. The edge graph of this polytope is planar, which allows us to define an appropriately small separator of the input point set.

For the $n^{\Omega(\sqrt{k})}$ lower bound, the first observation is that ORTHANT COVER for $d = 3$ is at least as hard as GEOMETRIC SET COVER in the plane where the objects are translates of an equilateral triangle. For the problem of GEOMETRIC SET COVER for squares, an $n^{\Omega(\sqrt{k})}$ conditional lower bound is known via a reduction from the GRID TILING problem [37, 40]. In this reduction, it is crucial that a gadget of (shifts of) a square can "transport" a value a from its left side to its right side, and a value b from its top side to its bottom side. For the related DOMINATING SET problem on intersection graphs of triangle translates, the proof strategy does generalize [18]. However, triangle translates are not flexible enough to naively follow this proof strategy for GEOMETRIC SET COVER: in a sense they have too few sides. Therefore, while our lower bound is also a reduction from GRID TILING, it is much more subtle. Our most crucial construction is a "sumcheck" gadget that obtains "input values" a and b at two sides of the involved triangles and results in the value $a+b$ at the "output side", while disallowing certain combinations. Using the "sumcheck" gadget on the values $a+b$ and $-a$ allows us to recover value b, and similarly we can recover a. Hence, we can use an essentially planar layout of triangles to transfer a value a from left to right and a value b from top to bottom; see Figure 3 for illustrations. Using this construction we can then simulate Grid Tiling, obtaining the claimed lower bound.

The $n^{\Omega(k)}$ lower bound for $d = 4$ as well as our results on approximation algorithms all follow by relatively simple reductions to or from known results.

Organization. We prove Theorem 2 in Section 2, which implies Theorem 1.1 as per [46, Lemma 2.3] or [12, Section A.3]. We prove the remainder of Theorem 1 in Sections 3 and 4, which respectively contain a sketch of the lower bound for $d = 3$ and the lower bound for $d = 4$. Details of the lower bound for $d = 3$ as well as the proof of Theorem 3 are deferred to the full version of the paper.

2 Half-spaces in dimension 3

In this section, we prove Theorem 2 (and by extension, Theorem 1.1) by giving an algorithm for Geometric Set Cover for half-spaces in $\mathbb{R}^3$. An instance of this problem consists of a set of half-spaces $\mathcal{D}$ in $\mathbb{R}^3$, a set of points U in $\mathbb{R}^3$, and an integer k. The question is whether one can select k half-spaces from $\mathcal{D}$ so that every point of U is covered by at least one of them.

We shall say that a set of half-spaces $\mathcal{D}$ in $\mathbb{R}^3$ is *in general position* if no two boundaries of half-spaces in $\mathcal{D}$ are parallel, and no four boundaries of half-spaces in $\mathcal{D}$ meet at one point. Note that given an instance $(\mathcal{D}, U, k)$ of Geometric Set Cover for half-spaces, one may slightly perturb the half-spaces of $\mathcal{D}$ so that every half-space still covers the same subset of points in U as before, but after the perturbation they are in general position. Hence, we shall assume this property in all the considered instances of Geometric Set Cover for half-spaces.

2.1 Algorithm

Our algorithm will rely on the following balanced separator lemma.

▶ **Lemma 4.** *Suppose $(\mathcal{D}, U, k)$ is an instance of* Geometric Set Cover *for half-spaces in $\mathbb{R}^3$ where $\mathcal{D}$ is in general position, and let $\mathcal{S} \subseteq \mathcal{D}$ be an optimum solution to this instance, whose size ℓ satisfies $4 < \ell \leqslant k$. Then there exists a subset $\mathcal{S}_0 \subseteq \mathcal{S}$ with $|\mathcal{S}_0| \leqslant \mathcal{O}(\sqrt{k})$ and a partition $\mathcal{P}$ of $U \setminus \bigcup \mathcal{S}_0$ with $|\mathcal{P}| \leqslant k$, such that the following property holds: For each $W \in \mathcal{P}$, if ℓ_W is the optimum size of a solution to the instance $(\mathcal{D}, W, k)$, then $\ell_W \leqslant \frac{2}{3}\ell$ and $|\mathcal{S}_0| + \sum_{W\in\mathcal{P}} \ell_W \leqslant \ell$.*

Moreover, given $(\mathcal{D}, U, k)$, one can in time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot |U|^{\mathcal{O}(1)}$ enumerate a family $\mathcal{N}$ consisting of at most $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})}$ pairs $(\mathcal{S}_0, \mathcal{P})$ with $\mathcal{S}_0 \subseteq \mathcal{D}$, $\mathcal{P}$ a partition of $U \setminus \bigcup \mathcal{S}_0$ with $|\mathcal{P}| \leqslant k$, and the guarantee that $\mathcal{N}$ contains at least one pair satisfying the property above.

Before we give a proof of Lemma 4, we show how it can be used to construct an algorithm as promised in Theorem 2. The algorithm is presented below using pseudo-code as Algorithm `halfSpaceCoverDim3`.

Algorithm 1 Algorithm `halfSpaceCoverDim3`.

Input: Instance $(\mathcal{D}, U, k)$ of Geometric Set Cover for half-spaces in $\mathbb{R}^3$ with $\mathcal{D}$ in general position
Output: An optimum solution $\mathcal{S} \subseteq \mathcal{D}$ provided it has size $\leqslant k$, or $\bot$ otherwise

```
S ← ⊥
for each C ⊆ D with |C| ≤ min(k, 4) do
    if U ⊆ ⋃C and |C| < |S| then                // convention: |⊥| = ∞
        S ← C
if k ≤ 4 then
    return S
N ← family enumerated using the algorithm of Lemma 4 for (D, U, k)
for each (S_0, P) ∈ N do
    for each W ∈ P do
        S_W ← halfSpaceCoverDim3(D, W, ⌊2k/3⌋)
    C ← S_0 ∪ ⋃_{W∈P} S_W                       // convention: ⊥ ∪ X = ⊥
    if |C| ≤ k and |C| < |S| then
        S ← C
return S
```

As argued, we may assume that $\mathcal{D}$ is in general position. First, we look through all candidates $\mathcal{C}$ for a solution with $|\mathcal{C}| \leqslant 4$. In case any such $\mathcal{C}$ covering U is found, we store the smallest one as the optimum solution. Next, provided $k > 4$, we apply the algorithm of Lemma 4 to the instance $(\mathcal{D}, U, k)$ and enumerate a suitable family of pairs $\mathcal{N}$. For each $(\mathcal{S}_0, \mathcal{P}) \in \mathcal{N}$ we apply the algorithm recursively to all instances $(\mathcal{D}, W, \lfloor \frac{2}{3}k \rfloor)$ for $W \in \mathcal{P}$, yielding solutions $\mathcal{S}_W$. We then consider $\mathcal{C} = \mathcal{S}_0 \cup \bigcup_{W \in \mathcal{P}} \mathcal{S}_W$ as a candidate solution, provided none of $\mathcal{S}_W$ is equal to $\bot$. Finally, we output the smallest candidate solution of size at most k found.

The correctness of the algorithm follows immediately from Lemma 4. Indeed, if $(\mathcal{D}, U, k)$ admits a solution of size at most 4, then an optimum solution will be found in the initial search. Otherwise, Lemma 4 ensures us that for some pair $(\mathcal{S}_0, \mathcal{P}) \in \mathcal{N}$, the recursive calls of the algorithm will find solutions $\mathcal{S}_W$ for $W \in \mathcal{P}$ which together with $\mathcal{S}_0$ form an optimum solution to $(\mathcal{D}, U, k)$.

We are left with bounding the time complexity of the algorithm. Let $C > 0$ be such that the algorithm of Lemma 4 always returns a family $\mathcal{N}$ satisfying $|\mathcal{N}| \leqslant |\mathcal{D}|^{C\sqrt{k}}$. Let $T[d, k]$ be the maximum number of leaves of the recursion tree produced by the algorithm when applied to an instance with $|\mathcal{D}| = d$ and parameter k. Then $T[d, k] = 1$ for $k \leqslant 4$, while for $k > 4$ we have the following recursive inequality:

$$T[d, k] \leqslant k \cdot d^{C\sqrt{k}} \cdot T[d, \lfloor 2k/3 \rfloor].$$

Here, factor $k \cdot d^{C\sqrt{k}}$ comes from the fact that for at most $d^{C\sqrt{k}}$ pairs $(\mathcal{S}_0, \mathcal{P}) \in \mathcal{N}$ we apply the algorithm recursively to $|\mathcal{P}| \leqslant k$ instances with parameter $\lfloor 2k/3 \rfloor$. Unraveling the recursion, we have

$$T[d, k] \leqslant k^{\log_{3/2} k} \cdot d^{C\sqrt{k} \cdot (1 + (2/3)^{1/2} + (2/3)^{2/2} + (2/3)^{3/2} + \ldots)} = k^{\log_{3/2} k} \cdot d^{C'\sqrt{k}} = d^{\mathcal{O}(\sqrt{k})},$$

where $C' = C \cdot \frac{1}{1 - (2/3)^{1/2}}$.

We conclude that the recursion tree for an instance with $d = |\mathcal{D}|$ and parameter k has at most $d^{\mathcal{O}(\sqrt{k})}$ leaves, so it also has $d^{\mathcal{O}(\sqrt{k})}$ nodes. The internal computation for each node takes time $d^{\mathcal{O}(\sqrt{k})} \cdot |U|^{\mathcal{O}(1)}$, so the total running time of $d^{\mathcal{O}(\sqrt{k})} \cdot |U|^{\mathcal{O}(1)}$ follows.

2.2 Balanced separator lemma

We now move to the proof of Lemma 4, which spans the remainder of this section.

Since $\mathcal{S}$ is an optimum solution to $(\mathcal{D}, U, k)$, we have that $\mathcal{S}$ is *minimal* in the following sense: there is no $S \in \mathcal{S}$ such that $S \subseteq \bigcup_{T \in \mathcal{S}\setminus\{S\}} T$. It turns out that this minimality condition together with the assumption $|\mathcal{S}| > 4$ implies that $\mathcal{S}$ cannot cover the whole space; this is implied by the following result.

▶ **Theorem 5** (Danzer et al. [17]). *If a set of half-spaces $\mathcal{S}$ in $\mathbb{R}^3$ is minimal and $\bigcup \mathcal{S} = \mathbb{R}^3$, then $|\mathcal{S}| \leqslant 4$.*

For every half-space $S \in \mathcal{S}$ we may choose an affine function $\varphi_S \colon \mathbb{R}^3 \to \mathbb{R}$ so that

$$S = \{x \in \mathbb{R}^3 \colon \varphi_S(x) \leqslant 0\}.$$

In particular, we set $\varphi_S(x) = \langle x - v_S, n_S\rangle$, where v_S is a point in the boundary of S, the vector n_S is the normal of the boundary plane of S pointing away from S, and $\langle ., . \rangle$ denotes the inner product in $\mathbb{R}^3$. Let $\overline{S} = \{x \in \mathbb{R}^3 \colon \varphi_S(x) \geqslant 0\}$; that is, $\overline{S}$ is the closure of the complement of S. Then the complement of $\bigcup \mathcal{S}$ is the interior of the polytope P defined as follows:

$$P = \bigcap_{S \in \mathcal{S}} \overline{S} = \{x \in \mathbb{R}^3 \colon \varphi_S(x) \geqslant 0 \text{ for all } S \in \mathcal{S}\}.$$

By Theorem 5 we infer that P is non-empty.

We shall also assume from now on that the polytope P is bounded. This can be achieved by adding to $\mathcal{S}$ up to 6 *dummy* half-spaces of the form $\{(x_1, x_2, x_3) \in \mathbb{R}^3 \colon x_i \geqslant M\}$ and $\{(x_1, x_2, x_3) \in \mathbb{R}^3 \colon x_i \leqslant -M\}$ for $i = 1, 2, 3$ and some large M, so that none of the dummy half-spaces covers any point of U. These may be perturbed slightly so that $\mathcal{S}$ remains in general position. As we will not use the optimality of $\mathcal{S}$ from now on, this can be safely done at the cost of replacing ℓ with $\ell + 6$ in all asymptotic bounds. Note that we do ensure that minimality of $\mathcal{S}$ is maintained, and thus possibly less than 6 dummy half-spaces are added.

Recall that we denote $|\mathcal{S}| = \ell$. Thus, P is a bounded convex polytope in $\mathbb{R}^3$ with ℓ faces, one for each half-space of $\mathcal{S}$ (this follows by minimality). Since $\mathcal{S}$ is in general position, at each vertex of P three faces meet. Let H be a graph whose vertices are the vertices of P and whose edges are the edges of P. Observe that the boundary of P – which consists of its faces – is homeomorphic to a sphere, so this homeomorphism shows that H admits a drawing in the sphere with ℓ faces. In the following, we identify faces of H with the faces of P. Since every face f of P is a polygon, the boundary of f is a simple cycle in H. Therefore, H is a simple 3-regular plane graph (i.e. without loops and multiple edges connecting the same pair of vertices) that is connected and bridgeless.

Let H' be the *radial graph* of H: the vertex set of H' consists of vertices and faces of H, and in H' a vertex u is adjacent to a face f if and only if u lies on the boundary of f. Note that H' is bipartite, with the vertices and faces of H being the bipartition. Also, H' admits an embedding into a sphere constructed from the embedding of H as follows: for every face f pick an arbitrary point $x_f \in f$ representing it, and connect x_f with all vertices u lying on f using pairwise non-crossing curves within f. Observe that every face of H' is a 4-cycle, induced by two faces of H and the endpoints of an edge shared by them. Since H is connected and bridgeless, a straightforward argument shows that H' is 2-connected. Since H is 3-regular, it follows that $3|V(H)| = 2|E(H)|$, so by Euler's formula for polyhedra ($|V(H)| - |E(H)| + \ell = 2$), we have that $|V(H)| = 2\ell - 4$. Consequently, $|V(H')| = 3\ell - 4$.

We may now apply the following Cycle Separator Theorem of Miller.

▶ **Theorem 6** ([42], with simplified formulation). *Let G be a 2-connected plane graph on n vertices and let d be the maximum length of a face in G. Suppose $\mu\colon V(G) \to [0,1]$ is a weight function on the vertices of G such that $\mu(V(G)) = \sum_{v \in V(G)} \mu(v) = 1$. Then there exists a simple cycle C in G of length at most $2\sqrt{2\lfloor d/2 \rfloor n}$ such that if R_1 and R_2 are the (open) connected regions of the plane with C removed, then the vertices contained in R_1 have total weight at most $2/3$, and the same holds for R_2.*

On the vertex set of H' define the following weight function: $\mu(f) = \frac{1}{\ell}$ for every face f of H, and $\mu(u) = 0$ for every vertex u of H. By Theorem 6, in H' there exists a simple cycle C of length at most $4\sqrt{|V(H')|} = 4\sqrt{3\ell - 4}$ such that every connected component of $H' - C$ contains at most $\frac{2}{3}\ell$ vertices that correspond to faces of H. Let

$$C = (z_1, f_1, z_2, f_2, \ldots, z_q, f_q),$$

where $2q$ is the length of C (thus $q \leqslant 2\sqrt{3\ell - 4}$), $z_1, \ldots, z_q$ are consecutive vertices of H visited by C, and $f_1, \ldots, f_q$ are consecutive faces of H visited by C.

Let Q be a closed poly-line in $\mathbb{R}^3$ with vertices $z_1, \ldots, z_q$, connected with straight line segments in this order (cyclically). Then the segment between z_i and z_{i+1} (with indices behaving cyclically modulo q) is entirely contained in the face f_i of P. Thus, Q is a curve contained in the boundary of P (denoted further ∂P), so it splits ∂P (which is homeomorphic to a sphere) into two regions, say A_1 and A_2.

We now color the faces of P in three colors as follows:

- faces incident to any of the vertices $z_1, z_2, \ldots, z_q$ are colored green;
- remaining faces are colored red or blue, depending whether they are contained in A_1 or A_2.

Note that since three faces meet at each vertex z_i, there are at most $4\sqrt{3\ell - 4}$ green faces: $f_1, \ldots, f_q$ and at most one additional face per each vertex z_i. Also, red faces do not share edges with blue faces, because all faces intersecting Q (even at one point) are colored green. We treat the above coloring of faces of P also as a coloring of all the points of ∂P. Here, points on edges of P are colored green if any face incident to the edge is colored green, and they are colored red or blue if all incident faces are red or blue, respectively.

Let $X = \operatorname{conv}\{z_1, \ldots, z_q\}$. The asserted properties of C immediately yield the following.

▷ Claim 7. There are at most $4\sqrt{3\ell - 4}$ green faces, at most $\frac{2}{3}\ell$ red faces, and at most $\frac{2}{3}\ell$ blue faces. No red face shares any edge with any blue face. Moreover, if x is any blue point on ∂P and y is any red point on ∂P, then the straight line segment with endpoints x and y intersects X.

As faces of P are in one-to-one correspondence with the half-spaces of $\mathcal{S}$, we may talk about red, green, and blue half-spaces of $\mathcal{S}$. We next observe that the separating properties of C carry over to the points of U.

▷ Claim 8. If a point $u \in U$ is simultaneously covered by a red half-space from $\mathcal{S}$ and by a blue half-space from $\mathcal{S}$, then it is also covered by a green half-space from $\mathcal{S}$.

Proof. Let A and B be respectively the red and the blue half-space covering u, and let f_A and f_B be the faces of P that correspond to A and B, respectively. Pick any point $x_A \in f_A$ and $x_B \in f_B$ and let Π be a plane through u, x_A, and x_B. We may choose x_A and x_B so that Π does not contain any vertex of P. Then $P \cap \Pi$ is a nonempty convex polygon, whose sides are colored red, green, and blue so that no red side is adjacent to any blue side. Moreover, the side containing x_A is red, while the side containing x_B is blue. Call these sides s_A and s_B, respectively.

Call a side s of $P \cap \Pi$ *separating* if its extension to a line separates u from $P \cap \Pi$ in the plane Π. Since $P \cap \Pi$ is convex, separating sides form an interval on the perimeter of $P \cap \Pi$. Moreover, since A and B cover u, it follows that s_A and s_B are separating. As s_A is red and s_B is blue, from the two claims above we conclude that there exists a green side s of $P \cap \Pi$ that is also separating. Then the half-space corresponding to the face of P containing s is green and it covers u, as required. ◁

By Claim 8, we may partition U into three subsets:

- *green points* of U that are covered by some green half-space from $\mathcal{S}$;
- *red points* of U that are covered only by red half-spaces from $\mathcal{S}$;
- *blue points* of U that are covered only by blue half-spaces from $\mathcal{S}$.

Denote the above sets by U_G, U_R, U_B, respectively. In the following claims, roughly speaking we show that X can be used to separate red points of U from blue points of U. Call two points $u, v \in U$ *separated* by X if the straight line segment with endpoints u and v intersects X.

▷ Claim 9. For all $u \in U_R$ and $v \in U_B$, we have that u and v are separated by X.

Proof. Let I be the straight line segment with endpoints u and v.

Suppose first that I does not intersect the polytope P. Since both I and P are convex, there exists an affine functional $\psi \colon \mathbb{R}^3 \to \mathbb{R}$ such that $\psi(u) < 0$, $\psi(v) < 0$, but $\psi(x) \geqslant 0$ for all $x \in P$. We may moreover choose ψ so that there exists a vertex w of P for which $\psi(w) = 0$. Let the faces of P incident to w be contained in the boundaries of half-spaces S_1, S_2, S_3. Since ψ is nonnegative on P, it follows that ψ can be written as a nonnegative linear combination of $\varphi_{S_1}, \varphi_{S_2}, \varphi_{S_3}$. Then $\varphi_{S_i}(u) < 0$ holds for some index $i \in \{1, 2, 3\}$, and similarly condition $\varphi_{S_j}(v) < 0$ holds for some index $j \in \{1, 2, 3\}$. Thus S_i covers u and S_j covers v, so S_i is necessarily red and S_j is necessarily blue. However, the faces corresponding to S_i and S_j share an edge incident to the vertex w. This contradicts Claim 7.

Now we know that I indeed intersects P. Let $J = I \cap P$. Note that since $u \in U_R$, the endpoint of J closer to u has to be red, for the half-space corresponding to the face of P containing this endpoint covers u. Similarly, the endpoint of J closer to v has to be blue. We conclude that, by Claim 7, the segment J has to intersect X. ◁

▷ Claim 10. Suppose $u, v \in U \setminus U_G$ are separated by X. Then there is no half-space in $\mathcal{S}$ that would simultaneously cover both u and v.

Proof. Let I be the straight line segment with endpoints u and v, and let x be any point of $I \cap X$.

Suppose first that x lies on ∂P. Since $x \in X$ and all faces of P incident to $z_1, \ldots, z_q$ are colored green, it follows that x is green. Let S be any half-space of $\mathcal{S}$ corresponding to a green face on which x lies. As $x \in I$, we conclude that S either covers u or v, which contradicts the assumption that $u, v \notin U_G$.

Suppose now that x lies in the interior of P. If there was a half-space $S \in \mathcal{S}$ containing both u and v, then S would contain the whole segment I, and x in particular, so S would intersect the interior of P. This is a contradiction with the definition of P. ◁

Consider now a graph L with vertex set $U \setminus U_G$, where different $u, v \in U \setminus U_G$ are considered adjacent if and only if they are not separated by X. Then Claims 9 and 10 directly imply the following.

▷ Claim 11. Every connected component of L is entirely contained either in U_R or in U_B. Moreover, no half-space in $\mathcal{S}$ covers points from two different connected components of L.

The existential part of Lemma 4 follows now if we take $\mathcal{S}_0$ to be the set of green half-spaces and $\mathcal{P}$ to be the partition of $U \setminus U_G$ into the connected components of L. Here, if any half-space from $\mathcal{S}_0$ turns out to be one of the at most six dummy half-spaces, we may safely remove it from $\mathcal{S}_0$, as it does not cover any point in U anyway. Let us check that the required properties are indeed satisfied by the pair $(\mathcal{S}_0, \mathcal{P})$. First, by Claim 7 we have $|\mathcal{S}_0| \leqslant 2q \leqslant \mathcal{O}(\sqrt{k})$. Next, for a connected component W of L, let us denote by $\mathcal{S}_W$ the set of half-spaces from $\mathcal{S}$ that cover at least one point of $U \setminus U_G$ belonging to W. Clearly, $\mathcal{S}_W$ is a solution to the instance $(\mathcal{D}, W, k)$, hence $|\mathcal{S}_W| \geqslant \ell_W$. By Claim 10, the sets $\mathcal{S}_W$ are pairwise disjoint, and they are clearly disjoint from $\mathcal{S}_0$. Hence, we have

$$\ell = |\mathcal{S}| \geqslant |\mathcal{S}_0| + \sum_{W \in \mathsf{cc}(L)} |\mathcal{S}_W| \geqslant |\mathcal{S}_0| + \sum_{W \in \mathsf{cc}(L)} \ell_W,$$

where $\mathsf{cc}(L)$ is the set of connected components of L. Also, the sets $\mathcal{S}_W$ are non-empty, because every connected component of L requires at least one half-space to be covered, so $\sum_{W \in \mathsf{cc}(L)} |\mathcal{S}_W| \leqslant \ell$ entails that $|\mathcal{P}| = |\mathsf{cc}(L)| \leqslant \ell \leqslant k$. Finally, by Claim 9, for each $W \in \mathsf{cc}(L)$ the half-spaces of $\mathcal{S}_W$ are either all red or all blue, which by Claim 7 implies that $|\mathcal{S}_W| \leqslant \frac{2}{3}\ell$ for all $W \in \mathsf{cc}(L)$.

We are left with providing an algorithm enumerating a suitable family $\mathcal{N}$. The algorithm proceeds as follows. Let $\mathcal{D}'$ be $\mathcal{D}$ augmented by adding the six dummy half-spaces of the form $\{(x_1, x_2, x_3) \in \mathbb{R}^3 \colon x_i \geqslant M\}$ and $\{(x_1, x_2, x_3) \in \mathbb{R}^3 \colon x_i \leqslant -M\}$ for $i = 1, 2, 3$ and some large M, so that none of the added half-spaces covers any point of U. Say that a point $x \in \mathbb{R}^3$ is *important* if it is the intersection of some triple of planes that are boundaries of some half-spaces in $\mathcal{D}'$. Note that all vertices of the polytope P are important points, while the total number of important points is at most $(|\mathcal{D}| + 6)^3$ and they can be enumerated in time $\mathcal{O}(|\mathcal{D}|^3)$.

Next, for every $q \leqslant 2\sqrt{3 \cdot (k+6) - 4} = 2\sqrt{3k+14}$, iterate through

- every choice of $2q$ half-spaces from $\mathcal{D}$, say $\mathcal{S}_0 = \{S_1, \ldots, S_{2q}\}$;
- and every choice of q important points $z_1, \ldots, z_q$.

Note that there are at most $|\mathcal{D}|^{4\sqrt{3k+14}}$ choices for $\mathcal{S}_0$ and at most $(|\mathcal{D}| + 6)^{6\sqrt{3k+14}}$ choices for $z_1, \ldots, z_q$, hence we iterate through at most $(|\mathcal{D}| + 6)^{10\sqrt{3k+14}}$ choices in total.

Let $X = \operatorname{conv}\{z_1, \ldots, z_q\}$ and let U_G be the set of all points of U that are covered by some half-space of $\mathcal{S}_0$. Construct the graph L as described before: the vertex set of L is $U \setminus U_G = U \setminus \bigcup \mathcal{S}_0$, and two points $u, v \in U \setminus U_G$ are adjacent if and only if u and v are not separated by X. Observe that whether u and v are separated by X can be checked in strongly polynomial time. Indeed, this question boils down to the verifying whether, in 3-dimensional Euclidean space, a given segment intersects a polyhedron defined as the convex hull of a given set of points, which can be solved by any strongly polynomial-time procedure for intersecting two convex polyhedra in $\mathbb{R}^3$, see e.g. [51, Section 7.3 and notes and comments to Chapter 7]. Therefore, L can be computed in (strongly) polynomial time. Finally, if L has at most k connected components, then include in the constructed family $\mathcal{N}$ the pair $(\mathcal{S}_0, \mathcal{P})$, where $\mathcal{P}$ is the partition of $U \setminus U_G$ into the connected components of L.

The bound on the size of $\mathcal{N}$ and the running time of the algorithm follow immediately from the description. The correctness is also clear, as some choice of $\mathcal{S}_0$ and $z_1, \ldots, z_q$ considered by the algorithm is the same as the one considered in the existential argument. This finishes the proof of Lemma 4.

3 Lower bound for dimension 3

The goal of this section is to prove Theorem 1, assertion 2. We can restrict our attention to the case where all points lie on the plane $\Pi = \{(x, y, z) \colon x + y + z = 0\}$ and the corners of all orthants lie on the plane $\{(x, y, z) \colon x + y + z = 1\}$. In such a setting, the intersections of the orthants with Π form equilateral triangles on Π, which all have the same size and orientation. In essence, this setting of Orthant Cover is equivalent to finding a geometric set cover of size k among m translates of some triangle. We call this problem Triangle Translate Cover. Therefore, Theorem 1.2 is implied by the following theorem.

▶ **Theorem 12.** *There is no $f(k)n^{o(\sqrt{k})}$ algorithm for* Triangle Translate Cover *for any computable function f, unless ETH fails.*

Here n is m plus the number of points.

Our reduction is from Grid Tiling [38, 16], which is defined as follows. We are given as input an integer k, an integer n, and a collection $\mathcal{S}$ of k^2 non-empty sets $S_{i,j} \subseteq \{1, \dots, n\} \times \{1, \dots, n\}$ for $i, j \in \{1, \dots, k\}$. The goal is to select an element $s_{i,j} \in S_{i,j}$ for each $i, j \in \{1, \dots, k\}$ such that:

- If $i < k$, $s_{i,j} = (x, y)$, and $s_{i+1,j} = (x', y')$, then $x = x'$.
- If $j < k$, $s_{i,j} = (x, y)$, and $s_{i,j+1} = (x', y')$, then $y = y'$.

One can picture these sets in a $k \times k$ matrix: in each cell (a, b), we need to select a representative from the set $S_{i,j}$ so that the representatives selected from horizontally neighboring cells agree in the first coordinate, and representatives from vertically neighboring cells agree in the second coordinate. Observe that due to equality conditions, the goal in the Grid Tiling problem can be stated equivalently as follows: select elements $x_1, \dots, x_k, y_1, \dots, y_k \in [n]$ such that $(x_i, y_j) \in S_{i,j}$ for all $i, j \in [k]$. Note that $s_{i,j} = (x_i, y_j)$ in this case. In the following, we will treat the selection $x_1, \dots, x_k, y_1, \dots, y_k$ also as a *solution* to a Grid Tiling instance.

Our goal is to create a parameterized reduction where the constructed instance of Triangle Translate Cover has a cover of size ck^2 for some constant c if and only if the original Grid Tiling instance has a solution. This is sufficient due to the following theorem.

▶ **Theorem 13** ([38, 16])**.** *There is no $f(k)n^{o(k)}$ algorithm for* Grid Tiling *for any computable function f, unless ETH fails.*

3.1 Gadgets

Due to lack of space, we only give a short intuitive overview of our construction. The complete construction and all proofs can be found in the full version of the paper.

Let $\varepsilon = \frac{1}{100n}$. In the Triangle Translate Cover problem, the input triangles are equilateral triangles; we assume the side lengths are precisely 1. This means our construction can effectively use three directions, namely along the vectors $\bar{\mathbf{e}} = (1, 0)$, $\acute{\mathbf{e}} = (1/2, \sqrt{3}/2)$, and $\grave{\mathbf{e}} = (1/2, -\sqrt{3}/2)$. For convenience, we let $E = \{\bar{\mathbf{e}}, -\bar{\mathbf{e}}, \acute{\mathbf{e}}, -\acute{\mathbf{e}}, \grave{\mathbf{e}}, -\grave{\mathbf{e}}\}$. Given a positive integer N, we use $[N]$ to denote $\{1, \dots, N\}$ and $[-N]$ to denote $\{-1, \dots, -N\}$.

Bundles. We first establish a gadget to represent an integer value. Let $N = 2n$ and $\mathbf{e} \in E$. A *bundle* is a set of N triangles $B = \{t_1, \dots, t_N\}$ such that t_1 has its lower-left corner on the origin and t_{i+1} is t_1 translated by $i\varepsilon \cdot \mathbf{e}$. The bundle also contains a point p_B on the incenter of $t_{N/2}$ ensuring that at least one triangle is selected from B. The idea behind the construction is that each solution will select exactly one triangle from the bundle. In this manner, the index of the selected triangle represents an integer in $[N]$. In the figures, each bundle has an arrow that indicates the direction along which the translation is done, and the indices (i.e. the represented integer) increase.

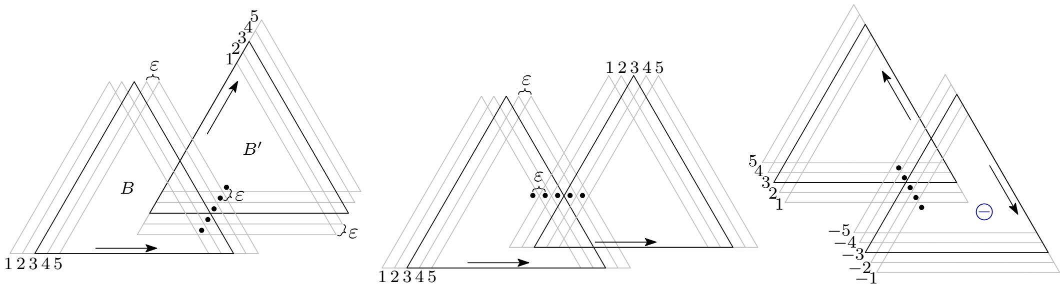

Figure 1 Left, middle: transportation gadgets. Arrows indicate the direction of increasing indices within the bundle. Right: negation gadget.

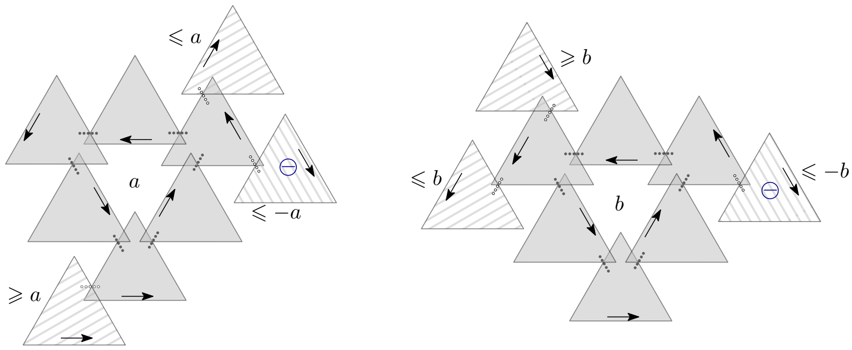

Figure 2 Integer gadgets with some transportation to the outside.

A *negative bundle* uses a different indexing, and represents the integer $-N+i-1$. Hence, the index of the selected triangle represents an integer in $[-N]$. (In figures, such bundles are indicated with a minus sign.)

Transportation gadget. We now establish a gadget to transport an integer value over some distance; this is built on a pair of bundles B and B' as in Figure 1. Observe that the boundaries of the triangles of B and B' induce an $(N-1) \times (N-1)$ lattice with directions **è** and **ē**. Now place points in the cells of this lattice as indicated in the figure. In this way, we are able to transport the integer value i represented by the triangle selected from B to an integer value of at most i for the bundle B'. Note that within certain limits, we can translate B' at will, so that we can "lengthen" or "shorten" as needed for the rest of the construction. By switching the sign of the values represented by one bundle of the transportation gadget, we get a *negation gadget* (see right hand side of Figure 1); if we make a cycle by joining transportation gadgets, we get an *integer gadget*, within which the selected triangles of each bundle must represent the same integer value (see Figure 2).

Sum and sumcheck gadget. We can create a *sum gadget*, which has three bundles, two of which are considered input bundles and one an output bundle. The gadget has the property that if the triangles selected in the two input bundles represent a and b respectively, then the output bundle must have a triangle representing some value that is at most $a+b$. Such a gadget is depicted in Figure 3. By adding extra points (indicated by red crosses), we can also disable the selection of certain triplets $(a, b, a+b)$. For a set $S \subseteq [n] \times [n]$, this allows us to create a *sumcheck gadget* where given inputs a, b the output is at most $a+b$, where equality can occur if and only if $(a, b) \in S$. This is the crucial step that eventually allows us to check for the sets $S_{i,j}$ corresponding to the cell i, j of the Grid Tiling instance.

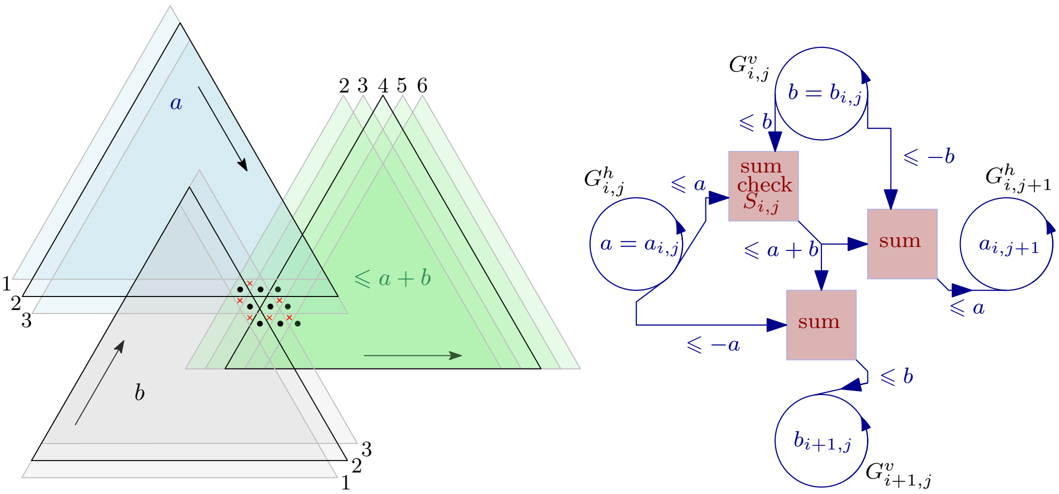

Figure 3 Left: summation gadget. Adding the points indicated by the red crosses creates an S-sumcheck gadget for $S = \{(1,1);(1,3);(2,2)\}$. Right: schematic representation of a cell (i,j) of the Grid Tiling instance. The transportation gadgets (blue) carry inequalities involving $a = a_{i,j}$ and $b = b_{i,j}$.

3.2 The complete construction

The idea of the reduction is to have for any pair of neighboring tiles an integer gadget which contains the value that these neighboring tiles must agree on. Given such integer gadgets, we can realize a single tile $(i,j) \in [k] \times [k]$ using our gadgetry the following way. We want to transfer the value a that is our horizontal selection and the value b which is our vertical selection between these integer gadgets. At the same time, we want to ensure that $(a,b) \in S_{i,j}$. We do this as explained schematically in Figure 3. From the integer gadgets on the left and on the top, we extract the integer values stored there, say $a_{i,j}$ and $b_{i,j}$ respectively, and transport these values (using transportation gadgets) to an $S_{i,j}$-sumcheck gadget. The output of this gadget will be an integer c satisfying $c \leqslant a_{i,j} + b_{i,j}$, and moreover $c < a_{i,j} + b_{i,j}$ if $(a_{i,j}, b_{i,j}) \notin S_{i,j}$. Using negation gadgets, we can extract $-a_{i,j}$ and $-b_{i,j}$ from the left and top integer gadgets, respectively. Each of these values can be combined with c through a sum gadget, whose output (i.e., third bundle) recovers integer values $a_{i,j+1} \leqslant a_{i,j}$ and $b_{i+1,j} \leqslant b_{i,j}$ that can be passed along to the right hand side and bottom integer gadgets respectively.

Let G be the construction thus far. Note that the construction ensures that left-to-right and top-to-bottom we have non-increasing values stored in our integer gadgets. To ensure equality holds, we need to wrap the rows and columns into cycles, just as we did for a single integer gadget. Doing this in a naive manner would lead to further crossings, so instead we create a construction H that is similar to G, but the rows are in reverse order, and the gadgetry of every tile is mirrored on the vertical axis; this construction is then translated below G (see Figure 4). In particular, the cell in row i and column j of H corresponds to the cell in row $k-i+1$ and column j of the Grid Tiling instance. As Figure 4 indicates, we can create transportation gadgets in a suitable manner to realize this construction.

We remark that it is tempting to use a known a variant of Grid Tiling called Grid Tiling with $\leqslant$ as the starting point of the reduction, which enjoys the same complexity lower bound as Grid Tiling; see [16, Theorem 14.30]. In this variant, the equality conditions are replaced with the requirement that one coordinate behaves non-decreasingly along rows, while the second behaves non-decreasingly along columns. The variant looks convenient, as our

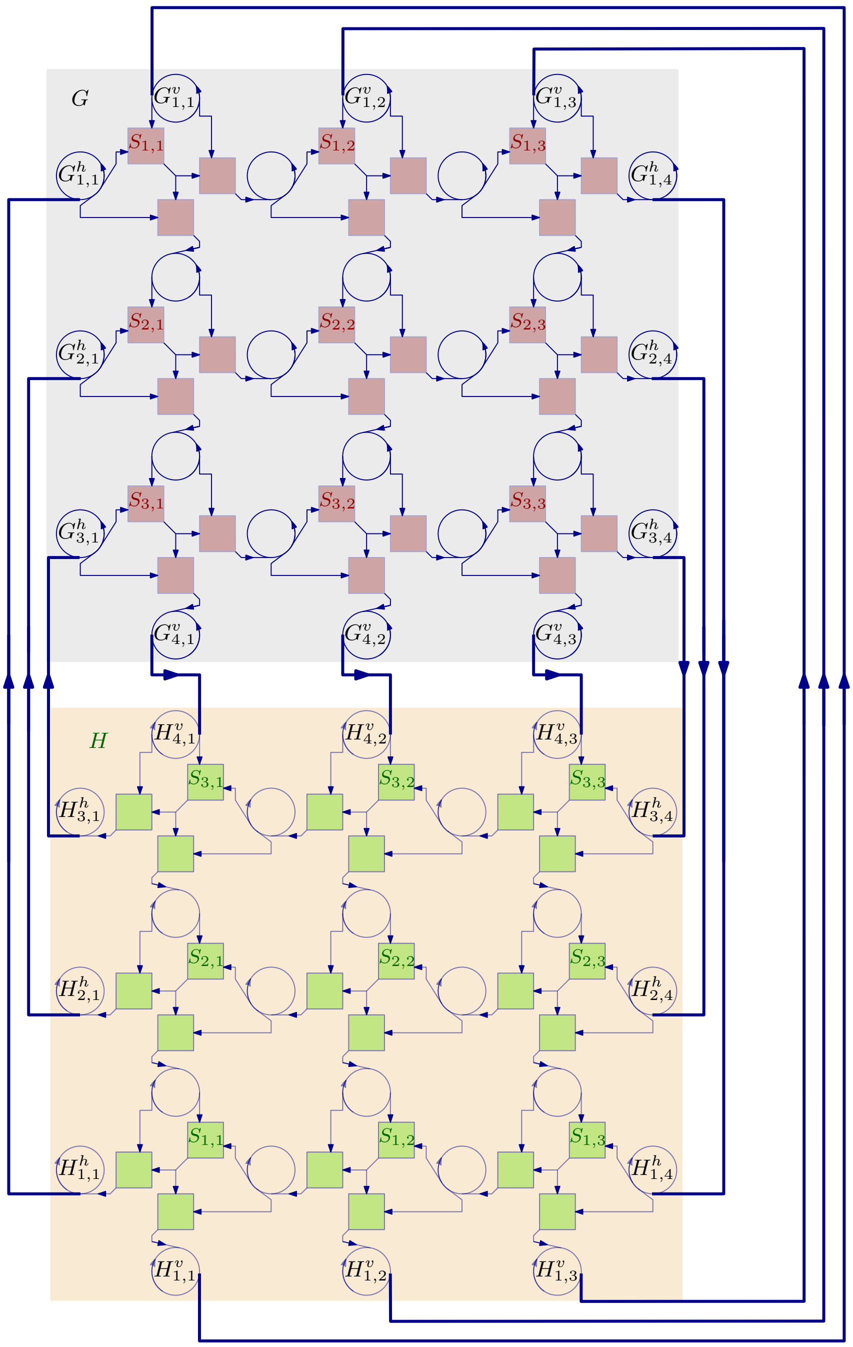

Figure 4 Schematic representation of the complete construction for $k = 3$.

gadgets directly implement inequalities between coordinates, not equalities. This thinking, however, is problematic for the following reason: in our construction, in order to implement the check $(a, b) \in S_{i,j}$ we have to enforce equality in the sumcheck gadget created for the cell (i, j), as in case of any slackness, this condition is not checked by the gadget. Therefore, starting the reduction from Grid Tiling with $\leqslant$ would not simplify the reasoning.

4 Lower bound for dimension 4 and higher

Consider the Rectangle Cover problem: Given points $P \subseteq \mathbb{R}^2$, a set R of axis-parallel rectangles in the plane, and a number k, decide whether there is a subset $R' \subseteq R$ of size k such that P is contained in the union of all rectangles in R'. Rectangle Cover is not solvable in time $f(k)n^{o(k)}$ for any computable f assuming ETH [39], where $n = |P| + |R|$. We obtain the same lower bound for Orthant Cover in dimension $d \geqslant 4$ by an easy reduction.

Proof of Theorem 1.3. Given points P and rectangles R in the plane, we construct a 4-dimensional Orthant Cover instance $(U, \mathcal{T})$: For each point $p = (x, y) \in P$, we add the point $(-x, x, -y, y)$ to U. For each rectangle $r = [x_1, x_2] \times [y_1, y_2] \in R$, we add the orthant with corner $(-x_1, x_2, -y_1, y_2)$ to $\mathcal{T}$. Note that p is contained in r if and only if $x_1 \leqslant x \leqslant x_2$ and $y_1 \leqslant y \leqslant y_2$, which is equivalent to $-x \leqslant -x_1$, $x \leqslant x_2$, $-y \leqslant -y_1$, and $y \leqslant y_2$. For points $p, q \in \mathbb{R}^d$, note that q is contained in the orthant $T = (-\infty, p_1] \times \ldots \times (-\infty, p_d]$ if and only if every coordinate of p is not larger than the corresponding coordinate of q. This proves the correctness of our reduction. We thus ruled out time $f(k)n^{o(k)}$ assuming ETH for Orthant Cover in dimension $d = 4$, and also for any $d \geqslant 4$ (by a trivial embedding). ◀

Together with the reasoning presented in [39], the above argument yields a chain of reductions from Clique to Orthant Cover in $d = 4$. Using recent hardness of approximation for Clique [9] and carefully tracking the gap through this chain of reductions, we obtain that Orthant Cover in $d = 4$ has no 1.05-approximation with running time $f(k)n^{o(k)}$ for any computable f, assuming Gap-ETH (Theorem 3.2). This result is presented in the full version of the paper.

References

1 Pankaj K. Agarwal and Jiangwei Pan. Near-Linear Algorithms for Geometric Hitting Sets and Set Covers. In Siu-Wing Cheng and Olivier Devillers, editors, *30th Annual Symposium on Computational Geometry, SOCG'14, Kyoto, Japan, June 08 - 11, 2014*, page 271. ACM, 2014.

2 Boris Aronov, Esther Ezra, and Micha Sharir. Small-Size ε-Nets for Axis-Parallel Rectangles and Boxes. *SIAM Journal on Computing*, 39(7):3248–3282, 2010.

3 Rom Aschner, Matthew J. Katz, Gila Morgenstern, and Yelena Yuditsky. Approximation Schemes for Covering and Packing. In Subir Kumar Ghosh and Takeshi Tokuyama, editors, *WALCOM: Algorithms and Computation, 7th International Workshop, WALCOM 2013, Kharagpur, India, February 14-16, 2013. Proceedings*, volume 7748 of *Lecture Notes in Computer Science*, pages 89–100. Springer, 2013.

4 Wolf-Tilo Balke, Ulrich Güntzer, and Jason Xin Zheng. Efficient Distributed Skylining for Web Information Systems. In *Advances in Database Technology, EDBT'04, 9th International Conference on Extending Database Technology*, volume 2992 of *LNCS*, pages 256–273. Springer, 2004.

5 René Beier and Berthold Vöcking. Random knapsack in expected polynomial time. *J. Comput. Syst. Sci.*, 69(3):306–329, 2004.

6 Karl Bringmann, Tobias Friedrich, and Patrick Klitzke. Generic Postprocessing via Subset Selection for Hypervolume and Epsilon-Indicator. In *13th International Conference on Parallel Problem Solving from Nature, PPSN XIII*, volume 8672 of *LNCS*, pages 518–527. Springer, 2014.

7 Karl Bringmann, Tobias Friedrich, and Patrick Klitzke. Two-dimensional subset selection for hypervolume and epsilon-indicator. In *Genetic and Evolutionary Computation Conference, GECCO'14*, pages 589–596. ACM, 2014.

8 Hervé Brönnimann and Michael T. Goodrich. Almost Optimal Set Covers in Finite VC-Dimension. *Discrete and Computational Geometry*, 14(1):463–479, 1995. doi:10.1007/BF02570718.

9 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From Gap-ETH to FPT-Inapproximability: Clique, Dominating Set, and More. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS'17*, pages 743–754, 2017.

10 Timothy M. Chan and Elyot Grant. Exact algorithms and APX-hardness results for geometric packing and covering problems. *Comput. Geom.*, 47(2):112–124, 2014.

11 Timothy M. Chan, Elyot Grant, Jochen Könemann, and Malcolm Sharpe. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1576–1585. SIAM, 2012.

12 Timothy M. Chan, Kasper Green Larsen, and Mihai Patrascu. Orthogonal Range Searching on the RAM, Revisited. *CoRR*, abs/1103.5510, 2011. arXiv:1103.5510.

13 Vasek Chvátal. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. doi:10.1287/moor.4.3.233.

14 Kenneth L. Clarkson and Kasturi R. Varadarajan. Improved Approximation Algorithms for Geometric Set Cover. *Discrete & Computational Geometry*, 37(1):43–58, 2007.

15 Vincent Cohen-Addad, Arnaud de Mesmay, Eva Rotenberg, and Alan Roytman. The Bane of Low-Dimensionality Clustering. In *29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'18*, pages 441–456. SIAM, 2018.

16 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

17 Ludwig Danzer, Branko Grünbaum, and Viktor Klee. Helly's theorem and its relatives. In *Proceedings of Symposia in Pure Mathematics*, volume 7, pages 101–180, 1963.

18 Mark de Berg, Sándor Kisfaludi-Bak, and Gerhard Woeginger. The complexity of Dominating Set in geometric intersection graphs. *Theoretical Computer Science*, 769:18–31, 2019. doi:10.1016/j.tcs.2018.10.007.

19 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016. URL: http://eccc.hpi-web.de/report/2016/128.

20 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633. ACM, 2014.

21 Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness. *Congressus Numerantium*, 87:161–178, 1992.

22 Thomas Erlebach, Hans Kellerer, and Ulrich Pferschy. Approximating Multiobjective Knapsack Problems. *Management Science*, 48(12):1603–1612, 2002.

23 Thomas Erlebach and Erik Jan van Leeuwen. PTAS for weighted set cover on unit squares. In Maria J. Serna, Ronen Shaltiel, Klaus Jansen, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings*, volume 6302 of *Lecture Notes in Computer Science*, pages 166–177. Springer, 2010.

24 Robert J. Fowler, Mike S. Paterson, and Steven L. Tanimoto. Optimal Packing and Covering in the Plane are NP-Complete. *Information Processing Letters*, 12(3):133–137, 1981. doi:10.1016/0020-0190(81)90111-3.

25 Sathish Govindarajan, Rajiv Raman, Saurabh Ray, and Aniket Basu Roy. Packing and Covering with Non-Piercing Regions. In Piotr Sankowski and Christos Zaroliagis, editors, *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 47:1–47:17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

26 Pierre Hansen. Bicriterion path problems. In *Multiple Criteria Decision Making Theory and Application*, pages 109–127. Springer, 1980.

27 Sariel Har-Peled. Being Fat and Friendly is Not Enough. *CoRR*, abs/0908.2369, 2009. arXiv:0908.2369.

28 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

29 David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, December 1974. doi:10.1016/S0022-0000(74)80044-9.

30 David S. Johnson. The NP-Completeness Column: An Ongoing Guide. *Journal of Algorithms*, 3(2):182–195, 1982. doi:10.1016/0196-6774(82)90018-9.

31 Sándor Kisfaludi-Bak, Jesper Nederlof, and Erik Jan van Leeuwen. Nearly ETH-tight algorithms for Planar Steiner Tree with terminals on Few Faces. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'19*, 2019. to appear.

32 Vladlen Koltun and Christos H. Papadimitriou. Approximately dominating representatives. *Theoretical Computer Science*, 371(3):148–154, 2007.

33 Sören Laue. Geometric Set Cover and Hitting Sets for Polytopes in $\mathbb{R}^3$. In Susanne Albers and Pascal Weil, editors, *STACS 2008, 25th Annual Symposium on Theoretical Aspects of Computer Science, Bordeaux, France, February 21-23, 2008, Proceedings*, volume 08001 of *Dagstuhl Seminar Series*, pages 479–490. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2008. URL: http://drops.dagstuhl.de/opus/volltexte/2008/1367.

34 Lászlo Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4):383–390, 1975. doi:10.1016/0012-365X(75)90058-8.

35 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In *44th International Colloquium on Automata, Languages, and Programming, ICALP'17*, volume 80 of *LIPIcs*, pages 78:1–78:15, 2017.

36 Dániel Marx. Parameterized Complexity of Independence and Domination on Geometric Graphs. In *Parameterized and Exact Computation, Second International Workshop, IWPEC, Proceedings*, pages 154–165, 2006. doi:10.1007/11847250_14.

37 Dániel Marx. On the Optimality of Planar and Geometric Approximation Schemes. In *48th Annual IEEE Symposium on Foundations of Computer Science, FOCS'07*, pages 338–348, 2007.

38 Dániel Marx. A Tight Lower Bound for Planar Multiway Cut with Fixed Number of Terminals. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 677–688. Springer, 2012.

39 Dániel Marx and Michał Pilipczuk. Optimal Parameterized Algorithms for Planar Facility Location Problems Using Voronoi Diagrams. In *ESA 2015*, volume 9294 of *Lecture Notes in Computer Science*, pages 865–877. Springer, 2015. See arXiv:1504.05476 for the full version.

40 Dániel Marx and Anastasios Sidiropoulos. The limited blessing of low dimensionality: when $1 - 1/d$ is the best possible exponent for d-dimensional geometric problems. In *30th Annual Symposium on Computational Geometry, SOCG'14*, page 67. ACM, 2014.

41 Jiří Matoušek, Raimund Seidel, and Emo Welzl. How to Net a Lot with Little: Small epsilon-Nets for Disks and Halfspaces. In Raimund Seidel, editor, *Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkeley, CA, USA, June 6-8, 1990*, pages 16–22. ACM, 1990.

42 Gary L. Miller. Finding Small Simple Cycle Separators for 2-Connected Planar Graphs. *J. Comput. Syst. Sci.*, 32(3):265–279, 1986.

43 Nabil H. Mustafa, Rajiv Raman, and Saurabh Ray. Quasi-Polynomial Time Approximation Scheme for Weighted Geometric Set Cover on Pseudodisks and Halfspaces. *SIAM J. Comput.*, 44(6):1650–1669, 2015.

44 Nabil H. Mustafa and Saurabh Ray. Improved Results on Geometric Hitting Set Problems. *Discrete & Computational Geometry*, 44(4):883–895, 2010. doi:10.1007/s00454-010-9285-9.

45 Frank Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research*, 181(3):1620–1629, 2007.

46 János Pach and Gábor Tardos. Tight lower bounds for the size of epsilon-nets. In *Symposium on Computational Geometry*, pages 458–463. ACM, 2011.

47 János Pach and Gerhard J. Woeginger. Some New Bounds for Epsilon-Nets. In Raimund Seidel, editor, *Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkeley, CA, USA, June 6-8, 1990*, pages 10–15. ACM, 1990.

48 Christos H. Papadimitriou and Mihalis Yannakakis. On the Approximability of Trade-offs and Optimal Access of Web Sources. In *41st Annual Symposium on Foundations of Computer Science, FOCS'00*, pages 86–92. IEEE Computer Society, 2000.

49 Christos H. Papadimitriou and Mihalis Yannakakis. Multiobjective Query Optimization. In *20th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS'01*. ACM, 2001.

50 Mihai Pătraşcu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, pages 1065–1075. SIAM, 2010.

51 Franco P. Preparata and Michael Ian Shamos. *Computational Geometry — An Introduction*. Texts and Monographs in Computer Science. Springer, 1985.

52 Anastasios Sidiropoulos, Kritika Singhal, and Vijay Sridhar. Fractal Dimension and Lower Bounds for Geometric Problems. In *34th International Symposium on Computational Geometry, SoCG'18*, volume 99 of *LIPIcs*, pages 70:1–70:14, 2018.

53 Warren D. Smith and Nicholas C. Wormald. Geometric Separator Theorem & Applications. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 232–243. IEEE, 1998.

54 Erik Jan van Leeuwen. *Optimization and Approximation on Systems of Geometric Objects*. PhD thesis, University of Amsterdam, 2009.

55 Kasturi R. Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 641–648. ACM, 2010.

56 Sergei Vassilvitskii and Mihalis Yannakakis. Efficiently computing succinct trade-off curves. *Theoretical Computer Science*, 348(2-3):334–356, 2005.

57 Daniel Vaz, Luís Paquete, and Aníbal Ponte. A note on the ε-indicator subset selection. *Theoretical Computer Science*, 499:113–116, 2013.